

- 1 1. (Currently Amended) A first computer system comprising:  
2 at least one processor;  
3 a memory coupled to the at least one processor;  
4 a datastream residing in the memory and received from a second computer system  
5 coupled to the first computer system, the datastream including:  
6 a send mechanism that is invoked by the second computer system to send  
7 the datastream to the first computer system; and  
8 a receive mechanism that is invoked by the first computer system to  
9 receive the datastream;  
10 a datastream factory residing in the memory and executed by the at least one  
11 processor, the datastream factory creating an instance of a datastream class corresponding  
12 to an identifier in [a] the datastream received from [a] the second computer system; and  
13 a datastream receive mechanism residing in the memory and executed by the at  
14 least one processor, the datastream receive mechanism defined in the datastream class,  
15 the datastream receive mechanism invoking the receive mechanism in the datastream to  
16 [causing] cause the instance of the datastream class to populate itself with information  
17 contained in the datastream [received from the second computer system by invoking at  
18 least one object method on the instance].
- 1 2. (Original) The computer system of claim 1 further comprising a datastream processing  
2 mechanism residing in the memory for processing the instance of the datastream by  
3 invoking at least one object method on the instance.
- 1 3. (Original) The computer system of claim 1 further comprising a datastream send  
2 mechanism residing in the memory for sending the instance of the datastream by invoking  
3 at least one object method on the instance.

1 4. (Original) The computer system of claim 1 wherein the datastream identifies  
2 executable code residing in the memory for receiving the datastream from the second  
3 computer system.

1 5. (Original) The computer system of claim 4 wherein the datastream further identifies  
2 executable code residing in the memory for performing a request represented by the  
3 datastream.

1 6. (Original) The computer system of claim 5 wherein the datastream further identifies  
2 executable code residing in the memory for sending the datastream from the second  
3 computer system to the computer system.

1 7. (Currently Amended) A networked computer system comprising:  
2 a first computer system coupled via a network connection to a second computer  
3 system;  
4 each of the first and second computer systems comprising a datastream processor,  
5 the datastream processor including:  
6 a datastream factory for creating an instance of an active datastream class  
7 corresponding to a datastream identifier received in a datastream on the network  
8 connection from the other computer system;  
9 a datastream send mechanism defined in the active datastream class that,  
10 when invoked, causes the datastream to be sent to the other computer system; and  
11 a datastream receive mechanism defined in the active datastream class  
12 that, when invoked, causes the instance of the active datastream class to populate  
13 itself with information contained in the datastream received on the network  
14 connection from the other computer system [by invoking at least one object  
15 method on the instance].

1 8. (Currently Amended) A networked computer system comprising:  
2 a first computer system coupled via a network connection to a second computer  
3 system;  
4 means for constructing an active datastream, the active datastream including a  
5 datastream identifier that identifies executable code for processing the active datastream;  
6 means within the active datastream for sending the active datastream from the first  
7 computer system to the second computer system;  
8 means for creating an instance of a datastream class that corresponds to the  
9 datastream identifier in the second computer system;  
10 means defined in the datastream class for causing the instance of the datastream  
11 class to populate itself with information contained in the active datastream received from  
12 the first computer system.

1 9. (Original) The computer system of claim 8 further comprising:  
2 means for processing the instance of the datastream class by invoking at least one  
3 object method on the instance.

1 10. (Original) The computer system of claim 8 further comprising:  
2 means for sending the instance of the datastream class by invoking at least one  
3 object method on the instance.

1 11. (Currently Amended) A method for communicating between a first computer system  
2 and a second computer system, the method comprising the steps of:  
3 the first computer system constructing an active datastream, the active datastream  
4 including:  
5 \_\_\_\_\_ a datastream identifier that identifies executable code for processing the  
6 active datastream;  
7 \_\_\_\_\_ a datastream send mechanism that is invoked by the first computer system  
8 to send the active datastream to the second computer system; and  
9 \_\_\_\_\_ a datastream receive mechanism that is invoked by the second computer  
10 system to receive the active datastream from the first computer system;  
11 the first computer system sending the active datastream to the second computer  
12 system by invoking the datastream send mechanism in the datastream;  
13 the second computer system creating an instance of a datastream class that  
14 corresponds to the datastream identifier; and  
15 the instance of the datastream class populating itself with information contained in  
16 the active datastream received from the first computer system by invoking the datastream  
17 receive mechanism [at least one object method on the instance].

1 12. (Original) The method of claim 11 further comprising the step of executing the  
2 executable code on the datastream instance to process the active datastream.

1 13. (Previously Presented) The method of claim 11 wherein the step of the instance of the  
2 datastream class populating itself with the information contained in the active datastream  
3 includes the step of executing a receive method on the instance of the datastream class.

1 14. (Original) The method of claim 11 wherein the step of the first computer system  
2 sending the active datastream to the second computer system includes the step of  
3 invoking at least one object method on the active datastream.

1 15. (Previously Presented) A method for communicating between a first computer system  
2 and a second computer system, the method comprising the steps of:  
3       the first computer system constructing an active datastream object, the active  
4 datastream object including a datastream identifier that identifies a corresponding  
5 datastream class that includes executable code corresponding to a plurality of object  
6 methods for processing the active datastream object;  
7       the first computer system sending the active datastream to the second computer  
8 system by invoking a send method on the active datastream object;  
9       the second computer system reading the datastream identifier from the active  
10 datastream object received from the first computer system;  
11       the second computer system creating a new instance of the datastream class that  
12 corresponds to the datastream identifier;  
13       the new instance of the datastream class populating itself with information  
14 contained in the active datastream received from the first computer system by invoking a  
15 receive method on the new instance; and  
16       the second computer system performing a request represented by the active  
17 datastream by invoking at least one object method on the new instance.

1 16. (Currently Amended) A program product comprising:  
2 a datastream factory that creates an instance of a datastream class corresponding  
3 to an identifier in a received datastream, the received datastream including:  
4 a send mechanism that is invoked to send the received datastream; and  
5 a receive mechanism;  
6 a datastream receive mechanism defined in the datastream class that invokes the  
7 receive mechanism to [causes] cause the instance of the datastream class to populate itself  
8 with information contained in the received datastream [by invoking at least one object  
9 method on the instance]; and  
10 signal bearing media bearing the datastream factory and the datastream receive  
11 mechanism.

1 17. (Original) The program product of claim 16 wherein the signal bearing media  
2 comprises recordable media.

1 18. (Original) The program product of claim 16 wherein the signal bearing media  
2 comprises transmission media.

1 19. (Original) The program product of claim 16 further comprising a datastream  
2 processing mechanism on the signal bearing media for processing the instance of the  
3 datastream by invoking at least one object method on the instance.

1 20. (Original) The program product of claim 16 further comprising a datastream send  
2 mechanism on the signal bearing media for sending the instance of the datastream by  
3 invoking at least one object method on the instance.

1 21. (New) A first computer system comprising:  
2 at least one processor;  
3 a memory coupled to the at least one processor;  
4 an active datastream object residing in the memory, the active datastream object  
5 comprising:  
6 a first object method invoked in the first computer system to send the  
7 active datastream object as a datastream to a second computer system coupled to  
8 the first computer system; and  
9 a second object method invoked in the second computer system that  
10 causes the second computer system to receive the datastream from the first  
11 computer system.

1 22. (New) The computer system of claim 21 further comprising:  
2 a datastream factory residing in the memory and executed by the at least one  
3 processor, the datastream factory creating an instance of a datastream class corresponding  
4 to an identifier the datastream; and  
5 a datastream receive mechanism defined in the datastream class that invokes the  
6 second object method to cause the instance of the datastream class to populate itself with  
7 information contained in the datastream.



1 23. (New) A method for communicating between a first computer system and a second  
2 computer system, the method comprising the steps of:  
3 the first computer system constructing an active datastream object, the active  
4 datastream object including:  
5 a first object method that is invoked to send the active datastream object as  
6 a datastream; and  
7 a second object method that is invoked to receive the datastream;  
8 the first computer system sending the datastream to the second computer system  
9 by invoking the first object method in the active datastream object;  
10 the second computer system creating an instance of a datastream class that  
11 corresponds to an identifier in the datastream; and  
12 the instance of the datastream class populating itself with information contained in  
13 the datastream received from the first computer system by invoking the second object  
14 method.

- 1 24. (New) A program product comprising:  
2 (A) an active datastream object comprising:  
3 a first object method invoked in a first computer system to send the active  
4 datastream object as a datastream to a second computer system coupled to the first  
5 computer system; and  
6 a second object method invoked in the second computer system that  
7 causes the second computer system to receive the datastream from the first  
8 computer system; and  
9 (B) computer-readable signal bearing media bearing the active datastream object.
- 1 25. (New) The program product of claim 24 wherein the signal bearing media comprises  
2 recordable media.
- 1 26. (New) The program product of claim 24 wherein the signal bearing media comprises  
2 transmission media.

## **STATUS OF THE CLAIMS**

Claims 1-20 were originally filed in this patent application. In response to the first office action dated 07/17/02, claims 1, 7, 8, 11, 13, 15 and 16 were amended in an amendment filed on 10/17/02. In response to the second office action dated 01/03/03, claims 1, 7, 8, 11, 13, 15 and 16 were amended in an amendment filed on 03/29/03. In response to the third office action dated 06/18/03, which was a final action, a Notice of Appeal was filed on 09/10/03, followed by an Appeal Brief that was filed on 11/05/03. In response to the Appeal Brief, the Examiner reopened prosecution in the pending office action. Claims 1-7 and 16-20 were rejected under 35 U.S.C. §103(a) as being unpatentable over De Borst in view of Henckel. Claims 8-14 were rejected under 35 U.S.C. §103(a) as being unpatentable over De Borst in view of Henckel and further in view of Susic. Claims 1-7 and 16-20 were rejected under 35 U.S.C. §103(a) as being unpatentable over De Borst in view of Dyer. Claims 8-14 were rejected under 35 U.S.C. §103(a) as being unpatentable over De Borst in view of Dyer and further in view of Susic. Claims 1-7 and 16-20 were rejected under 35 U.S.C. §103(a) as being unpatentable over De Borst in view of Nakano. Claims 8-14 were rejected under 35 U.S.C. §103(a) as being unpatentable over De Borst in view of Nakano and further in view of Susic. Claim 15 was allowed. In this amendment, claims 1, 7, 8, 11 and 16 have been amended, and new claims 21-26 have been added. Claims 1-26 are currently pending.